DOI: https://doi.org/10.52756/ijerr.2025.v47.008 *Int. J. Exp. Res. Rev.*, Vol. 47: 93-107 (2025)

Original Article | *Peer Reviewed* | *Open Access*

International Journal of Experimental Research and Review (IJERR)
© Copyright by International Academic Publishing House (IAPH)
ISSN: 2455-4855 (Online)
www.iaph.in
ISSN 2455-4855

Enhancing Cassava Disease Detection: Leveraging Deep Convolutional Neural Networks and Data Augmentation for Accurate Diagnosis

Poonam Chaudhary, Monika Lamba and Sneha K*

Data Science Lead, CSE Department, The NorthCap University, Gurugram, India

E-mail/Orcid Id:

PC, poonamchaudhary@ncuindia.edu, https://orcid.org/0000-0001-5529-5561; *ML,* monikalamba@ncuindia.edu, https://orcid.org/0000-0003-1695-9614; *SK,* snehakandacharam@ncuindia.edu, https://orcid.org/0009-0004-1330-1942

Article History:

Received: 08th Jul., 2024
Accepted: 25th Dec., 2024
Published: 30th Apr., 2025

Keywords:

Cassava disease detection, Deep learning architectures, Efficient Net, Disease surveillance

How to cite this Article:

Poonam Chaudhary, Monika Lamba and Sneha K (2025). Enhancing Cassava Disease Detection: Leveraging Deep Convolutional Neural Networks and Data Augmentation for Accurate Diagnosis. *International Journal of Experimental Research and Review*, *47*, 93-107.

DOI: https://doi.org/10.52756/ijerr.2025.v47.008

Abstract: Cassava, a widely cultivated staple food crop in the tropics, is frequently afflicted by various diseases that significantly decrease its yield. Cassava leaf disease diagnosis with four common cassava leaf diseases: cassava brown streak disease, cassava green mite, cassava bacterial blight, and cassava mosaic disease using state-of-the-art image classification techniques of deep CNN was implemented. In the proposed method data augmentation approaches have been applied to enlarge the dataset for better model performance. The methodology uses machine learning methods to surpass the conventional manual identification methods for recognizing affected leaf images, potentially increasing agricultural productivity through faster and more accurate disease detection that will lead to improved yield and better resource management in cassava cultivation. Evaluated the performance of various Deep Learning architectures, including LeNet, VGG-16, ResNet-50 and Efficient Net, as well as used various callback techniques. These include early stopping to prevent overfitting and model checkpointing, which saves the best-performing model, along with a learning rate reduction for fine-tuning these models and further optimizing their performance. The research, therefore, opens the possibilities of automated disease detection, greatly helping farmers in timely and accurate diagnosis for better management practices to ensure increased food security in cassava-growing regions, especially when mobile technology increases access to diagnostic tools and supports the field applicability of machine learning models. Further, deploying such models in the field has another advantage in embedding disease surveillance into the day-to-day activities of farmers, which again becomes an effective way to improve crop management strategies and responses in different agricultural contexts.

Introduction

Essential for guaranteeing food security in many poor countries, cassava—the third-largest source of dietary carbs in the tropics. However, the growth of cassava crops has been seriously endangered by numerous pests and diseases, including cassava green mite, cassava brown streak disease, cassava bacterial blight, and cassava mosaic disease (Mohanty et al., 2016; Wulder et al., 2021; Shukla et al., 2022). Farmers often lack the expertise and resources to accurately identify these diseases, leading to substantial economic losses.

To address this challenge, we have leveraged the advancements in deep learning and computer vision to develop an automated system for the cassava leaf disease classification (Kumar et al., 2020; Uke et al., 2021; Deep Convolution Neural Network Based solution for Detecting Plant Diseases, 2022). Our approach involves the use of various deep convolutional neural network architectures, such as LeNet, VGG-16, ResNet-50 and EfficientNet, to accurately identify the different disease types (Deep Convolution Neural Network Based solution for Detecting Plant Diseases, 2022; Pathak et al., 2022; Upadhyay et al., 2024). We have also employed data augmentation approaches to enhance the dataset and improve the models' performance (Mohanty et al., 2016).

In order to enhance training efficiency and avoid overfitting, we have also employed a number of callback strategies, such as early stopping, model checkpointing,

***Corresponding Author:** snehakandacharam@ncuindia.edu

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

and learning rate reduction (Liang et al., 2019). These methods have made it possible for us to identify and categorize cassava leaf diseases with a high degree of accuracy.

The results of this research demonstrate the potential of automated disease detection in assisting farmers with timely and accurate diagnosis, leading to better management practices and increased food security in cassava-growing regions.

Because it can be processed into a variety of products and is resistant to harsh weather, cassava (*Manihot esculenta* Cranz) is one of the most important staples and crops for food security in Africa (Mohanty et al., 2016). Nevertheless, crop diseases continue to be a significant problem, resulting in an estimated 12-23 million dollars worth of yield loss annually (Shukla et al., 2022; Prasad and Agniraj, 2024). Farmers are unable to identify the disease, so government agricultural experts must visit farms to visually assess the plants and determine whether they are sick or healthy. Because tracking the disease's progression is a labor-intensive process, it is difficult to treat because there are few people who can accurately identify the disease.

Through computer vision algorithms running on farmers' smartphones, low-cost methods of identifying cassava illness could assist in tracking disease progression and boost yields. It concentrated on correctly identifying the four most prevalent cassava illnesses, namely Cassava Brown Streak Disease (CBSD), Cassava Mosaic Disease (CMD), Cassava Bacterial Blight (CBB), and Cassava Green Mite (CGM). Only skilled professionals can diagnose cassava disease because the many strains exhibit remarkably similar symptoms, and each leaf can exhibit them.

Manually identifying diseased leaves is a time-consuming as well as labor-intensive process that has traditionally served as a primary method for detecting plant diseases and assessing agricultural damage. Various techniques exist to identify deformed leaves and other plant diseases. This study explores whether farmers can use current algorithms to diagnose cassava plant diseases effectively. Accurate damage assessment based on collected data is essential for making informed decisions about future recovery efforts. Internal factors, such as the durability of pavement materials against environmental changes like temperature fluctuations, heavy rainfall, soil fertilization, and soil-borne insects, play a role in plant health. External factors include proper plant care and effective management practices. The incidence of plant diseases is rapidly increasing, particularly in Bangladesh, as the country continues to develop.

Neural networks have been utilized in deep learning, an advanced form of machine learning, to mimic the operations of the human brain. Unlike traditional methods that rely on semantic properties for classification, deep learning can automatically extract information from visual patterns, as suggests (LeCun et al., 2015). The evolution of computers into a vital technology across various fields, including defense, medicine, agriculture, and engineering, is largely due to their capability to analyze multimedia data, like images captured by different computing systems. Research into applying machine learning and neural networks in agriculture has gained momentum, leading to new applications and continual improvements over existing methods. Our approach focuses on recognizing artificially extracted features to provide precise detection within a given dataset. For prediction problems involving image data and minimal pre-processing, Convolutional Neural Networks (CNNs) are often the best solution, particularly for classifying large-scale images. Given the challenges in selecting the right architecture, it is crucial to explore and document effective designs to support our research. This paper, however, emphasizes deep learning-based models and compares their accuracy rates rather than solely focusing on CNN research. Our proposed model will predict cassava plant diseases from leaf images, enabling the application of appropriate treatments and interventions for affected crops.

In this research paper, we have employed various deep learning architectures, including LeNet, VGG-16, ResNet-50, and EfficientNet, to classify cassava leaf diseases efficiently (Vardhan and Swetha, 2023; Shruthi et al., 2019).

It is impossible to exaggerate the significance of trees for both the environment and human life. Similar to people and animals, plants can get sick. Plant development can be hampered by a variety of diseases that affect vegetation. Precise detection of those chronic conditions is crucial for the identification and treatment of plant pathogens. This paper presents three deep learning methods that use a plant's leaf analysis to identify and categorize various plant diseases. The suggested method enhances the ability to identify diseases and categorize collected leaves with deformities. The way the model works is by classifying images into two groups: healthy and diseased. Furthermore, multiple processing layers that learn the data visualizations with distinct levels of abstraction comprise deep learning architectures. The most important part of developing any recognition system is gathering data sets. To label an image is to identify the specific subject that we will be

DOI: https://doi.org/10.52756/ijerr.2025.v47.008

searching for. In order to detect diseases, it is essential to train the algorithms using those images to identify the subjects. The deep learning model InceptionResNet-V2 was then used to detect diseases after we trained the ResNetV50 model on those images to identify the loss and accuracy from the images provided to it. To put it briefly, we are comparing the accuracy rate of three deep learning models on a large data set comprising 38 classes, which will enable the general public to identify abnormalities in plants. In order to improve the state of our farmers' livelihoods and the horticulture industry, it will also assist harvesters involved in agricultural labor in identifying the spread of their farmed crops.

Literature Review

Plant diseases can arise for a variety of reasons. Examples include nematodes, bacteria, mycoplasma, viruses, viroid, fungi, and parasitic blooming plants (Azeta et al., 2023). A technological method can be used to identify plant illnesses to prevent them. In essence, the method alerts farmers and cultivators to illnesses. There are numerous ways to demonstrate plant diseases. Several researchers have developed image processing and pattern identification algorithms for various applications of smart agriculture, such as sorting crops, fruits, and vegetables, and detecting illness in a field (Emmanuel et al., 2023; Demilie, 2024). They are not discussed in this brief review. Imaging methods include thermography, RGB imaging, fluorescence imaging, and hyperspectral imaging. Thus, there is no accepted method for identifying plant damage. The neural network model's performance has also decreased as a result of poor parameter and layer selection. To identify citrus diseases in images of fruit and leaves, we use a particular combination of layers and parameter settings in the suggested CNN model (Aruna et al., 2023). We also examined multiple CNN model iterations and compared the outcomes with the original study. We present a multilayer CNN model for accurately diagnosing citrus diseases from fruit and leaf images. According to several studies, visual processing tests are less accurate than absolute evaluation processes based on photographs. When evaluating the most recent deviation, the CLS Rater performed consistently better than human experts. The basic operation of several of these photo-built tools for diagnosing fungal infections of leaves is the same. To start, ill flowers' old and damaged tissue is removed using pre-processing techniques. Then, discriminating abilities are eliminated for additional research. Finally, skills are grouped according to a given activity using unsupervised cluster algorithms or supervised kind algorithms. Along with advancements in computer technology, several interactive devices have also progressed. Barbados has created and modeled numerous sets of image-based segmented algorithms to enhance crop or plant disease and its functional characteristics.

The goal of image recognition in agricultural applications is to recognize. As a result, acquiring an image is more challenging than it is in most circumstances. There will also be an effect from the network and camera communication required for image handling and recognition. As such, intelligent recognition is more difficult in a mountainous environment. This study aims to develop an algorithm in the Neural Network platform for the difficult environment of mountainous regions, as well as research models for diagnosing crop diseases and insect pests. Pedometers, accelerometers, and other cattle management tools were most commonly utilized as motion-tracking sensors to gather data on animals' daily activities. This kind of sensor has only been used in studies on animal welfare until recently. A much lower frequency of physical and developmental traits was seen in the animal production subcategory. We included factors such as back fat and muscle thickness, age, gender, biometric traits, metabolites, and heat stress. The last characteristic may have detrimental implications on animal health and product quality; however, measurements of back fat and muscle thickness can be used to determine carcass lean yield.

Research Problem

The identification and diagnosis of crop “diseases is a crucial task in agriculture, as it directly impacts crop yield & quality. Traditionally, disease diagnosis has relied on manual visual inspection by experts, which can be time-consuming”, subjective, and resource-intensive, particularly in remote regions (Jayaprakash and Balamurugan, 2020; Thakur et al., 2022). Automated disease detection systems can address these challenges by leveraging computer vision and machine learning techniques (Albattah et al., 2021; Li et al., 2023).

The design of such a system for cassava disease diagnosis, however, faces several key considerations. First, the system must be robust to the various environmental conditions and camera settings encountered in in-situ leaf imaging by farmers (Albattah et al., 2015). Second, the system must be able to accurately differentiate between multiple diseases with similar symptoms, as well as varying degrees of disease severity (Wu et al., 2021). Third, the system should be able to effectively leverage the large amounts of

DOI: https://doi.org/10.52756/ijerr.2025.v47.008

unlabelled data that are typically available, as annotating disease samples can be labor-intensive.

Scholars have investigated the application of both conventional "machine learning approaches, like support vector machines, and more sophisticated deep learning techniques to tackle these issues (Reichstein et al., 2019). In plant disease detection and classification tasks, deep learning models in particular have demonstrated promising results because they can automatically learn relevant features from the image data" without requiring manual feature engineering (Wu et al., 2021).

Leaf Symptoms of Cassava Diseases

The dataset includes four major cassava diseases, each presenting distinct symptoms on the leaves. Despite having clear symptomatic prototypes, field data reveal that a single plant can often be afflicted by multiple diseases, complicating diagnosis. Here are the primary symptoms associated with each of the four key cassava diseases:

#Cassava Mosaic Disease (CMD)

In sub-Saharan Africa, CMD is the most common cassava disease, exhibiting a variety of symptoms on the leaves. These symptoms include mottling, twisted and deformed leaflets, mosaic patterns, and a general decrease in the size of the plant and its leaves. Affected leaves have patches of the normal green color scattered with different shades of white and yellow. These chlorotic patches indicate a decrease in chlorophyll, which adversely affects photosynthesis and reduces crop yield.

#Cassava Brown Streak Disease (CBSD)

CBSD, currently the most severe cassava disease, is transmitted by whiteflies and through contaminated cuttings. It is particularly widespread in East Africa and other sub-Saharan African regions where cassava is cultivated. Yellow or necrotic vein banding on the leaves, which can enlarge and combine to form larger yellow patches, is the characteristic feature of CBSD. Even in the absence of tuber symptoms, leaf and stem symptoms may manifest. These include a decrease in the size of the roots and dark-brown necrotic patches inside the tubers.

#Cassava Bacterial Blight (CBB)

A dangerous bacterial illness called CBB prefers moist conditions for growth. The virulence of the bacterial strains causing the illness, seasonal variations, and environmental factors can all affect the symptoms of CBB. Black leaf spots, angular leaf spots, blights, early drying, and leaf loss from young leaves wilting under severe attack are common leaf symptoms.

#Cassava Green Mite (CGM)

White spots on leaves caused by CGM infestation can spread to cover the entire leaf, which causes the chlorophyll to be lost. Additionally, CGM-affected leaves may show striped symptoms that could be confused with CMD. Severe damage can cause the leaves to shrink, dry out, and fall off the plant, giving it a distinctive candlestick shape.

Dataset Description

This dataset from Kaggle contains over 21,000 files and about 6.19 GB of cassava leaf disease dataset, ready to be used for identifying diseases automatically. It encompasses the training and testing of image files of cassava leaves that may have characteristics of different diseases. Train.csv is an important file that gives the filenames with labels of diseases; another set of important files that come in the form are [train/test] _tfrecords, which constitute image files in TensorFlow Record format.

Work Plan and Methodology

Plan of work prior to anything else, we need to gather plant pictures from farmers or agricultural companies. In this context, our study has focused on using ML approaches, like supervised deep learning algorithms, to anticipate illnesses and pest crops. As a result, we give a summary of supervised learning techniques frequently used for image detection in this study. Using data collection and image processing methods, a disease detection model was created in an automated way.

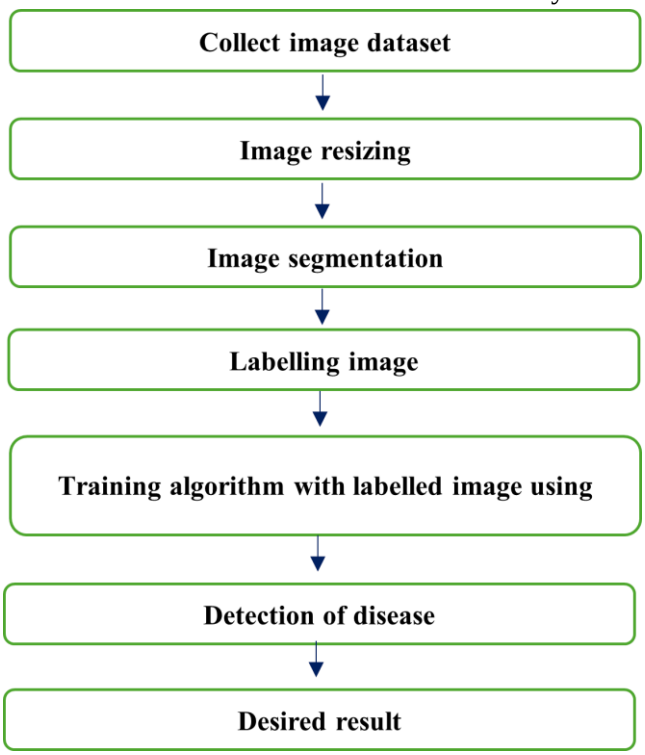


Figure 1. Workflow of Proposed Method.

DOI: https://doi.org/10.52756/ijerr.2025.v47.008

Image labeling is the process of identifying and highlighting particular features of an image. Using mage labeling with automated metadata creation is a bad idea. A very efficient deep-learning technique will be used to train the algorithm to identify diseases. After entering all of the picture data into the system, begin the process of photograph identification. Following that, image labeling software will be installed in order to tag these images using techniques that meet the predetermined standards.

Methodology

Inception Resnet V2 is used by the proposed plant disease detection model to identify diseases. To do this, the model must be created using a process that takes in data from an image data set, analyzes it efficiently, and generates predictions of two kinds: illnesses, or the study's objective was to assess the accuracy of diseases. It was a simple multi-class classification as a result. In our model, 38 classes were used. The methods of our suggested model are shown in a flowchart.

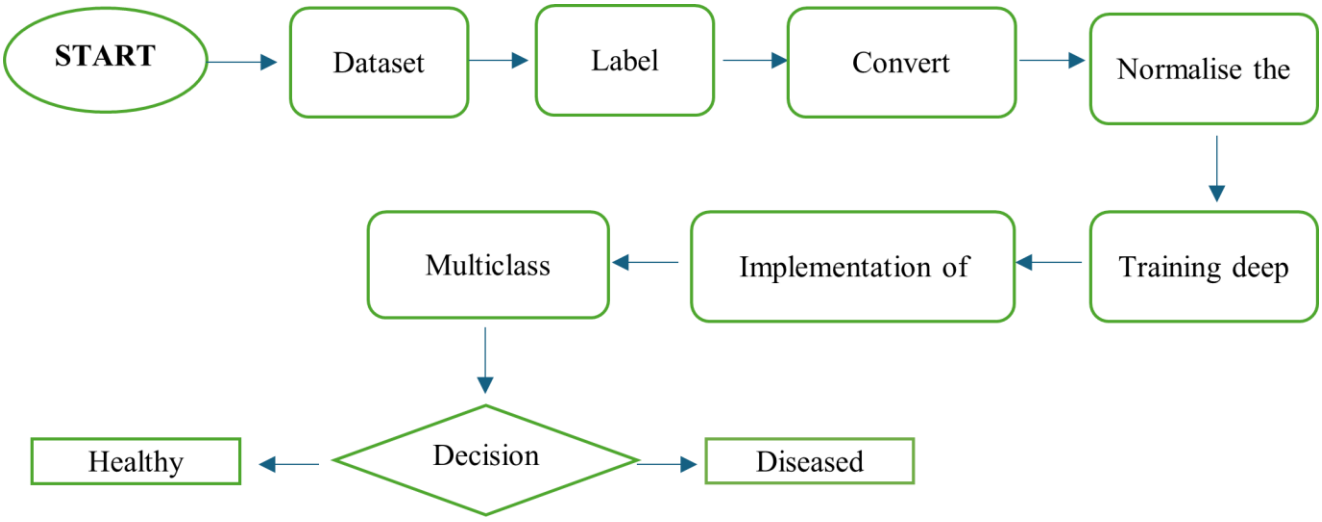


Figure 2. Flowchart of the suggested model.

The method for detecting diseases consists of three main steps:

1. Image pre-processing Data gathering and categorization of disease and non-disease incidences served as the study's initial step. The goal of this phase is to efficiently organize the input data, which consists of the whole collection of image data, for the illness detection algorithm.

2. Training: In this phase, the picture data set is transformed into arrays, the data set is normalized, and deep learning techniques (InceptionResnet V2) are used to train the model. Transfer learning and multi-class classification are also used to provide predictions.

3. Testing: After normalizing the data and dividing the population into two groups, this stage involves selecting random photos to test for the presence of any diseases.

Dataset Contextual Challenges

The dataset faces several challenges that complicate the identification of cassava diseases. These challenges include (1) varying image scales and backgrounds, (2) the time of day the images were captured, (3) the presence of multiple co-occurring diseases on the same plant, and (4) lack of focus in some images. These factors reflect real-world conditions, such as images taken in smallholder farmers' gardens where consistent backgrounds are unattainable. Attempts to standardize backgrounds with sheets were ultimately abandoned due to their impracticality.

Data Preparation for the Cassava Challenge

Providing the computer vision community with a dataset that accurately reflects the real-world challenges faced by many farmers was one of the main objectives of this challenge. As a result, during data processing, an attempt was made to preserve some of the previously mentioned contextual challenges. The four main steps of the data preparation pipeline were designed to guarantee the dataset's legitimacy and applicability to actual situations.

DOI: https://doi.org/10.52756/ijerr.2025.v47.008

Figure 3. Types of Cassava plants.

Methodology

LeNet:

In this research paper, we explore the architecture of the LeNet CNN for image classification tasks. LeNet, one of the earliest and most influential CNN architectures, has been widely used for various image recognition problems.

The dataset utilized in the research consists of 19,354 images, with 18,386 images for training and 968 images for validation. The images are divided into 5 classes, with 4 classes representing different diseases and 1 class representing healthy samples (Akash et al., 2023).

The preprocessing of the images involved converting them to grayscale with a size of 32x32 pixels.

The two convolutional layers in the LeNet architecture utilized in this work are separated by a max-pooling layer. Six 5x5 filters make up the first convolutional layer, and sixteen 5x5 filters make up the second. “The ReLU activation function is used by both convolutional layers.

After the convolutional and pooling layers, the model is flattened and fed into three fully connected dense layers. For multi-class classification, the output layer employs the Softmax activation function, while the first two dense layers use the ReLU activation” function.

To train the LeNet model, the "Adam" optimizer was used with a learning rate of 0.0002 and the loss function was set to "categorical cross-entropy."

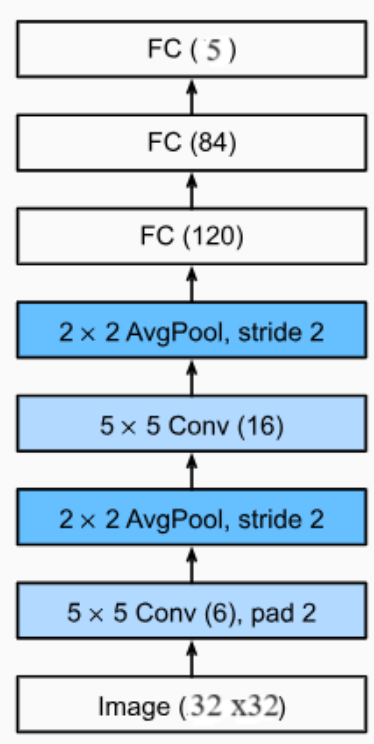


Figure 4. Architecture of LeNet.

DOI: https://doi.org/10.52756/ijerr.2025.v47.008

```
[ ] history = Model_fit(train_set, val_set)

Epoch 1/10
575/575 [==============================] - 141s 237ms/step - loss: 1.1922 - categorical_accuracy: 0.6048 - val_loss: 1.1822 - val_categorical_accuracy: 0.6116 - lr: 2.0000e-04
Epoch 2/10
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considered legacy.
  saving_api.save_model(
575/575 [==============================] - 107s 186ms/step - loss: 1.1633 - categorical_accuracy: 0.6119 - val_loss: 1.1962 - val_categorical_accuracy: 0.6116 - lr: 2.0000e-04
Epoch 3/10
575/575 [==============================] - 110s 192ms/step - loss: 1.1559 - categorical_accuracy: 0.6123 - val_loss: 1.1676 - val_categorical_accuracy: 0.6116 - lr: 2.0000e-04
Epoch 4/10
575/575 [==============================] - 105s 183ms/step - loss: 1.1429 - categorical_accuracy: 0.6141 - val_loss: 1.1354 - val_categorical_accuracy: 0.6126 - lr: 2.0000e-04
Epoch 5/10
575/575 [==============================] - 107s 186ms/step - loss: 1.1372 - categorical_accuracy: 0.6154 - val_loss: 1.1392 - val_categorical_accuracy: 0.6147 - lr: 2.0000e-04
Epoch 6/10
575/575 [==============================] - 106s 184ms/step - loss: 1.1292 - categorical_accuracy: 0.6173 - val_loss: 1.1399 - val_categorical_accuracy: 0.6167 - lr: 2.0000e-04
Epoch 7/10
575/575 [==============================] - 109s 189ms/step - loss: 1.1255 - categorical_accuracy: 0.6189 - val_loss: 1.1291 - val_categorical_accuracy: 0.6312 - lr: 2.0000e-04
Epoch 8/10
575/575 [==============================] - 109s 189ms/step - loss: 1.1245 - categorical_accuracy: 0.6194 - val_loss: 1.1279 - val_categorical_accuracy: 0.6136 - lr: 2.0000e-04
Epoch 9/10
575/575 [==============================] - 103s 179ms/step - loss: 1.1236 - categorical_accuracy: 0.6182 - val_loss: 1.1514 - val_categorical_accuracy: 0.6157 - lr: 2.0000e-04
Epoch 10/10
575/575 [==============================] - 106s 185ms/step - loss: 1.1235 - categorical_accuracy: 0.6180 - val_loss: 1.1229 - val_categorical_accuracy: 0.6147 - lr: 2.0000e-04
```

Figure 5. Screenshot of the implementation of the LeNet Model in Jupyter Notebook.

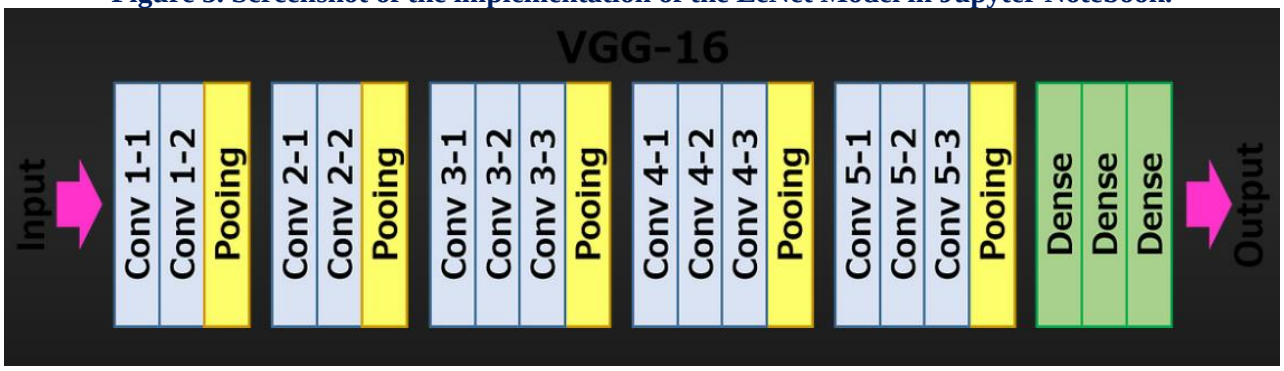


Figure 6. Architecture of VGG 16.

```
Epoch 2/10
636/636 [==============================] - 467s 735ms/step - loss: 1.1075 - accuracy: 0.6548 - val_loss: 1.6973 - val_accuracy: 0.6374
Epoch 3/10
636/636 [==============================] - 437s 688ms/step - loss: 1.1075 - accuracy: 0.6641 - val_loss: 1.1164 - val_accuracy: 0.6206
Epoch 4/10
636/636 [==============================] - 445s 699ms/step - loss: 1.1099 - accuracy: 0.6613 - val_loss: 0.9970 - val_accuracy: 0.6832
Epoch 5/10
636/636 [==============================] - 438s 689ms/step - loss: 1.1267 - accuracy: 0.6676 - val_loss: 1.2402 - val_accuracy: 0.6757
Epoch 6/10
636/636 [==============================] - 436s 686ms/step - loss: 1.1192 - accuracy: 0.6666 - val_loss: 1.1805 - val_accuracy: 0.6411
Epoch 7/10
636/636 [==============================] - 434s 683ms/step - loss: 1.1048 - accuracy: 0.6681 - val_loss: 1.1890 - val_accuracy: 0.6271
Epoch 8/10
636/636 [==============================] - 433s 681ms/step - loss: 1.1370 - accuracy: 0.6667 - val_loss: 1.3026 - val_accuracy: 0.6617
Epoch 9/10
636/636 [==============================] - 433s 680ms/step - loss: 1.1291 - accuracy: 0.6697 - val_loss: 1.2621 - val_accuracy: 0.6692
Epoch 10/10
636/636 [==============================] - 436s 686ms/step - loss: 1.1666 - accuracy: 0.6646 - val_loss: 1.0610 - val_accuracy: 0.6953
```

Figure 7. Implementation of the VGG16 model in Jupyter Notebook.

VGG 16

Here, we describe a thorough method for classifying plant diseases utilizing the VGG16 convolutional neural network architecture. The traditional manual diagnosis process for plant diseases can be labor-intensive and less accurate, highlighting the need for automated methods that can efficiently detect and categorize numerous plant diseases (Albattah et al., 2015). To tackle this challenge, we have leveraged the powerful feature extraction capabilities of the VGG16 model, which has demonstrated impressive performance in image classification tasks.

The first step in our methodology was to resize all input images to a uniform size of 224x224 pixels, as required by the VGG16 architecture. We then selected appropriate pre-trained weights for our dataset, which consisted of plant images with various disease conditions. Next, we imported the necessary libraries and the training dataset and used the Counter function to determine the frequency of each disease label. To further explore the dataset, we randomly selected 16 images and converted them to grayscale for visualization purposes.

DOI: https://doi.org/10.52756/ijerr.2025.v47.008

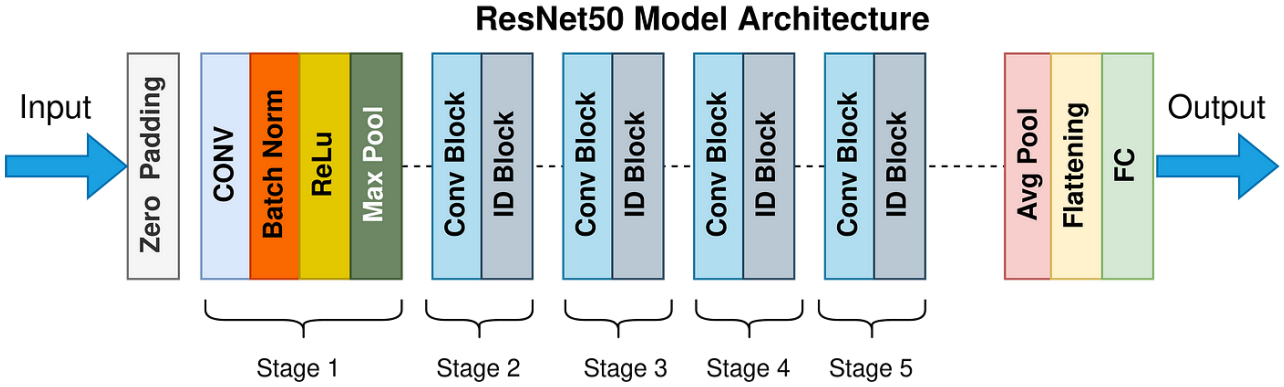


Figure 8. Architecture of Resnet 50.

```
Found 20327 validated image filenames belonging to 5 classes.
Found 1070 validated image filenames belonging to 5 classes.
Epoch 1/10
636/636 [==============================] - ETA: 0s - loss: 1.1960 - categorical_accuracy: 0.6130/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWa
  saving_api.save_model(
636/636 [==============================] - 341s 520ms/step - loss: 1.1960 - categorical_accuracy: 0.6130 - val_loss: 1.1712 - val_categorical_accuracy: 0.6150 - lr: 2.0000e-04
Epoch 2/10
636/636 [==============================] - 302s 475ms/step - loss: 1.1630 - categorical_accuracy: 0.6151 - val_loss: 1.1487 - val_categorical_accuracy: 0.6140 - lr: 2.0000e-04
Epoch 3/10
636/636 [==============================] - 299s 470ms/step - loss: 1.1573 - categorical_accuracy: 0.6153 - val_loss: 1.1443 - val_categorical_accuracy: 0.6150 - lr: 2.0000e-04
Epoch 4/10
636/636 [==============================] - 299s 471ms/step - loss: 1.1486 - categorical_accuracy: 0.6154 - val_loss: 1.1410 - val_categorical_accuracy: 0.6150 - lr: 2.0000e-04
Epoch 5/10
636/636 [==============================] - 302s 475ms/step - loss: 1.1386 - categorical_accuracy: 0.6159 - val_loss: 1.1305 - val_categorical_accuracy: 0.6150 - lr: 2.0000e-04
Epoch 6/10
636/636 [==============================] - 303s 476ms/step - loss: 1.1336 - categorical_accuracy: 0.6162 - val_loss: 1.1407 - val_categorical_accuracy: 0.6196 - lr: 2.0000e-04
Epoch 7/10
636/636 [==============================] - 305s 479ms/step - loss: 1.1305 - categorical_accuracy: 0.6171 - val_loss: 1.1328 - val_categorical_accuracy: 0.6234 - lr: 2.0000e-04
Epoch 8/10
636/636 [==============================] - 313s 492ms/step - loss: 1.1268 - categorical_accuracy: 0.6177 - val_loss: 1.1293 - val_categorical_accuracy: 0.6140 - lr: 2.0000e-04
Epoch 9/10
636/636 [==============================] - 307s 483ms/step - loss: 1.1263 - categorical_accuracy: 0.6176 - val_loss: 1.1191 - val_categorical_accuracy: 0.6196 - lr: 2.0000e-04
Epoch 10/10
636/636 [==============================] - 310s 487ms/step - loss: 1.1230 - categorical_accuracy: 0.6168 - val_loss: 1.1225 - val_categorical_accuracy: 0.6196 - lr: 2.0000e-04
```

Figure 9. Implementation of Resnet 50 in Jupyter Notebook.

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Figure 10. Architecture of EfficientNet.

Recognizing the importance of image preprocessing and normalization for effective model training, we performed these crucial steps on the dataset prior to feeding it to the VGG16 model.

The VGG16 model, with its 16 layers, was then loaded and a dense layer with a Softmax activation function was added to the end of the network.

RESNET-50

In our work with the Residual Network, we utilized the ResNet-50 architecture, specifically leveraging a pre-trained model without the top layer. The input shape was set to (180, 180 and 3), and we incorporated global average pooling, utilizing pre-trained weights from ImageNet. After that, a flattened layer was applied to create a one-dimensional output. This was followed by the addition of two dense layers, one with 512 units and the Softmax activation function for the other with 5 units.

We built the model with categorical cross-entropy as the loss function for training, which is suitable for tasks involving multiple classes of classification. We used a learning rate of 0.0002 for the Adam optimizer and tracked categorical accuracy as the evaluation metric. To enhance training efficiency, we implemented Early Stopping, Model Checkpoint, and ReduceLROnPlateau, all set to monitor the validation loss. Following that, we

DOI: https://doi.org/10.52756/ijerr.2025.v47.008

initiated the training process by fitting our model using the specified configuration and ran it for 10 epochs.

Efficient NET

Compared to more complex models like GPipe, these CNNs provide higher accuracy while greatly increasing model efficiency by drastically lowering parameters and FLOPS (Floating Point Operations Per Second). This network's central component is the optimized squeeze-and-excitation MBConv block. “Similar to the inverted residual blocks in MobileNet v2, MBConv creates a shortcut connection from a convolutional block's beginning to its end. The input activation maps are first expanded using 1x1 convolutions, which increases the feature maps' depth. Three-by-three depth-wise and point-wise convolutions come next, reducing the total number of channels in the final feature map. The wider layers are situated between these skip connections, while the shortcut connections connect the” narrower layers. This design minimizes the size of the model as well as the overall computational load.

In this deep convolutional neural network, we first imported leaf disease label files using a JSON file, which we converted into a dictionary named real_labels. This allowed us to create a mapping from numeric labels to the corresponding disease names. To balance our dataset, we used a sampling method to select a representative set of data from each class. These samples were stored in data frames named class_0 to class_4, which we then combined into a final data frame called final_data.

We then split the data into training and validation sets, specifying an image size of 300. To enhance model generalization, we applied data augmentation using various parameters of the ImageDataGenerator. We used a predefined model as the input for our "leaf model," which we compiled for training with the Adam optimizer at a learning rate of 0.0002, categorical accuracy as the evaluation metric, and categorical cross-entropy as the loss function—suitable for multi-class classification. To improve training, we implemented Early Stopping, Model Checkpoint, and ReduceLROnPlateau to monitor the learning process. Finally, we trained the model and stored the training history.

Research Objective

In this work, we use Inception-V2 deep learning models to develop a system that can recognize unhealthy plants. On the other hand, this system uses deep learning algorithms to collect plant leaf photos from any device and assess the health of the plant.

1. Build a data collection of images to train our algorithm for spotting abnormalities in a cassava plant's leaf so that it can produce a result right away.

2. To build the Transfer learning-based system, make sure that the different 38 classes of data are represented as images.

3. To avoid significant crop damage brought by pests, bacteria, or fungi.

Results and Discussion

Data collection

19,354 photos make up the dataset used in this study, of which 18,386 were used for training and 968 for validation. The images are divided into 5 classes, with 4 classes representing different diseases and 1 class representing healthy samples (Akash et al., 2023). The preprocessing of the images involved converting them to grayscale with a size of 32x32 pixels which are 38 distinct classes. This includes healthy, pest-affected, bacterial-infected, fungal-infected, nutrient-deficient, and environmentally stressed leaves. Each class here consists of a significant number of images to ensure the model learns the unique characteristics of each condition.

Building transfer learning system

After preprocessing, CNN models, such as LeNet, ResNet 50, VGG 16, and EfficientNet, were used. These models have learned rich feature representations that can be fine-tuned for cassava leaf abnormality detection.

```
Epoch 1/3
430/430 [==============================] - ETA: 0s - loss: 0.7046 - categorical_accuracy: 0.7738/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: User
  saving_api.save_model(
430/430 [==============================] - 694s 1s/step - loss: 0.7046 - categorical_accuracy: 0.7738 - val_loss: 0.4908 - val_categorical_accuracy: 0.8471 - lr: 2.0000e-04
Epoch 2/3
430/430 [==============================] - 622s 1s/step - loss: 0.6598 - categorical_accuracy: 0.7964 - val_loss: 0.5363 - val_categorical_accuracy: 0.8305 - lr: 2.0000e-04
Epoch 3/3
430/430 [==============================] - 624s 1s/step - loss: 0.6463 - categorical_accuracy: 0.7993 - val_loss: 0.4623 - val_categorical_accuracy: 0.8590 - lr: 2.0000e-04
<keras.src.callbacks.History at 0x7db2578c7520>
```

Figure 11. screenshot of jupyter notebook after training the model for 3 epochs, with each epoch taking approximately 30 minutes.

DOI: https://doi.org/10.52756/ijerr.2025.v47.008

import matplotlib.pyplot as plt
# Assuming you have already imported necessary libraries and defined
# Iterate over each test image and its corresponding prediction
for image_path, prediction in zip(test_images, predictions):
# Load and display the image
img = Image.open(image_path)
plt.imshow(img)
plt.axis('off')
# Set the title as the predicted label
plt.title(f'Predicted Label: {prediction}')
# Show the image with the predicted label
plt.show()
Predicted Label: 3

Figure 12. Screenshot of Prediction using LeNet model.

Figure 13. Screenshot of Prediction using VGG-16.

import matplotlib.pyplot as plt
# Assuming you have already imported necessary libraries and defined variables
# Iterate over each test image and its corresponding prediction
for image_path, prediction in zip(test_images, predictions):
# Load and display the image
img = Image.open(image_path)
plt.imshow(img)
plt.axis('off')
# Set the title as the predicted label
plt.title(f'Predicted Label: {prediction}')
# Show the image with the predicted label
plt.show()
Predicted Label: 0

Figure 14. Screenshot of Prediction using Resnet-50.

Figure 15. Screenshot of Prediction using EfficientNet Model.

DOI: https://doi.org/10.52756/ijerr.2025.v47.008

Training and validation (understanding which system helps in avoiding significant crop damage)

Both training and validation are important phases of coding a machine learning system. Training is the process by which the model finds patterns and features by changing its parameters in accordance with a labeled data set. It consists of supplying the model with input data files and measuring its prediction values against the right labels to decrease the percentage of errors through various ways like backpropagation. On the other hand, validation is a technique that helps us evaluate the model's performance on data that is yet to be seen by the model. This stage assists in fine-tuning the model, preventing overfitting, and ensuring it works well with other real-world data.

Training and Validation by four models are as follows:

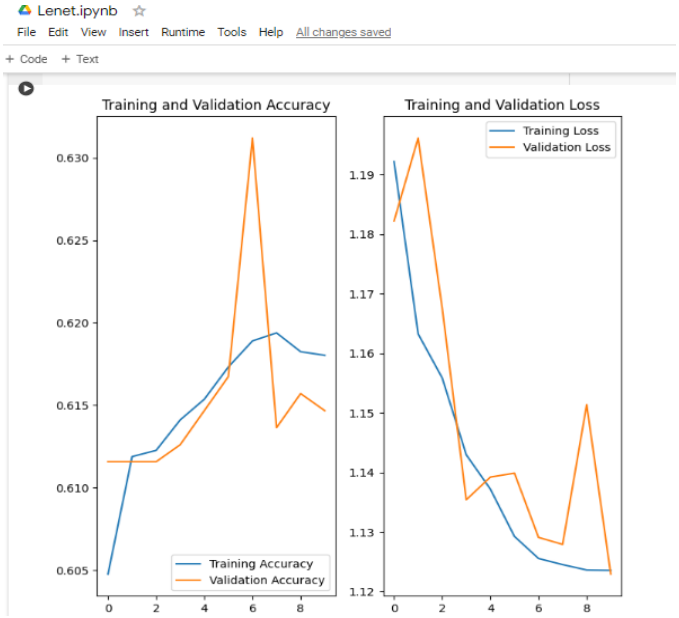
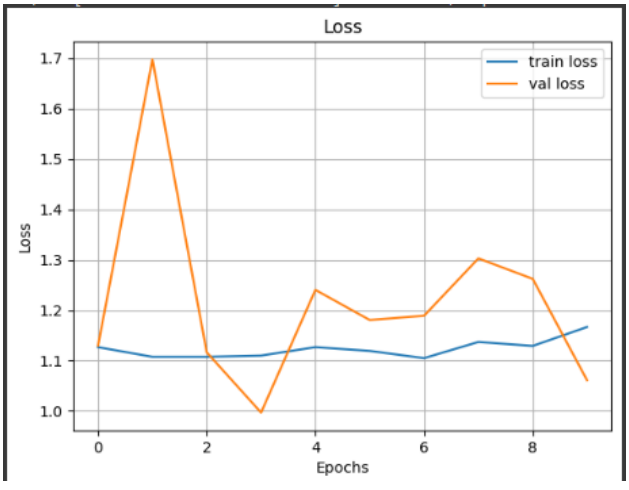


Figure 16. Screenshot of training and validation loss of LeNet Model.

Training achieved the best accuracy of 61.89%, while the validation accuracy peaked at 63.50%. The training loss was recorded at 1.125%, with a corresponding validation loss of 1.124%.



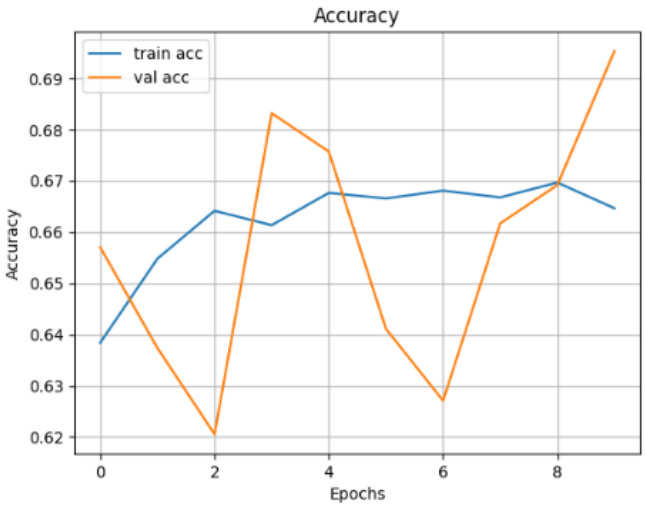


Figure 17. Screenshot of training and validation loss of VGG16 Model.

The training process reached a best accuracy of 66.46%, with the validation accuracy improving to 69.53%. The training loss was 1.1666%, and the validation loss was slightly lower at 1.0610%.

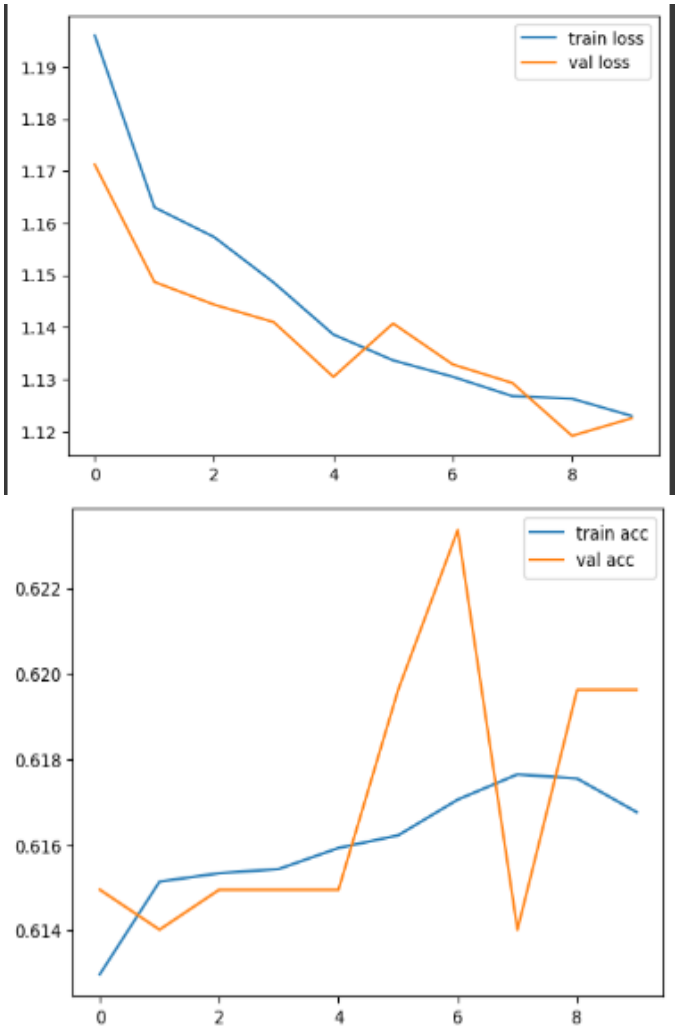


Figure 18. Screenshot of training and validation loss of Resnet-50 Model.

The training categorical accuracy was 0.6168%, with the validation categorical accuracy slightly higher at

DOI: https://doi.org/10.52756/ijerr.2025.v47.008

0.6196%. The training loss stood at 1.230%, while the validation loss was 1.225%.

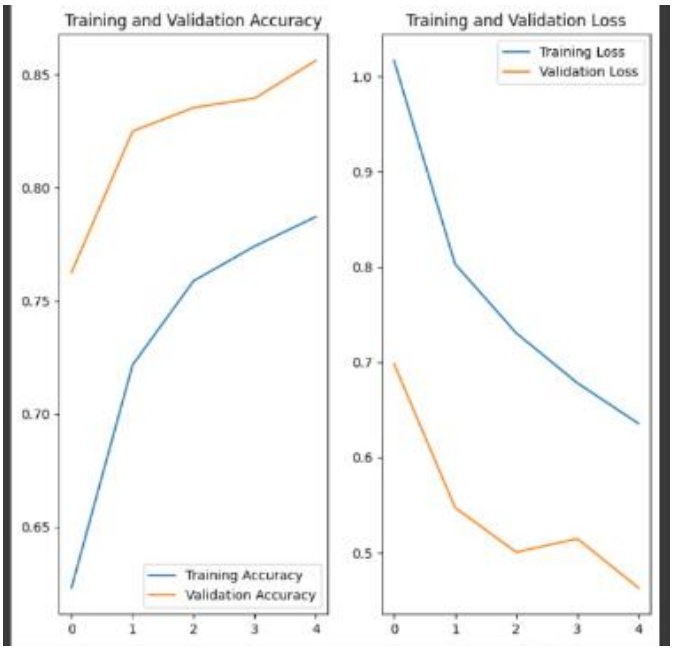


Figure 19. Screenshot of training and validation loss of Efficient Net Model.

The training process achieved a best accuracy of 75.93%, with the validation accuracy reaching 85.90%. The training loss was recorded at 0.6463% and the validation loss was lower at 0.4623%.

The dataset comprises 19,354 cassava leaf images, with 18,386 images used for training and 968 for validation. Images belong to five classes: four disease classes (CBSD, CMD, CBB, CGM) and one healthy class. Different lighting conditions, angles, and backgrounds were used while capturing images to reflect realistic farming environments. Data augmentation techniques such as rotation, flipping, and scaling improved the robustness of the models.

Deep Learning Architectures

We tried four CNN architectures: LeNet, VGG-16, ResNet-50, and EfficientNet. We trained and fine-tuned each of them on the cassava dataset.

LeNet: This is a light model and used mostly as a baseline.

VGG-16: It has 16 layers. The characteristic feature of this architecture is good feature extraction capability.

ResNet-50: In this architecture, the residual connections ensure that the deeper layers do not have the problem of vanishing gradients.

EfficientNet: Efficient model that could balance both depth, width, and resolution. Has the focus of achieving greater accuracy without using many parameters.

Model Training and Optimization

Each model is trained by a learning rate of 0.0002 along with categorical cross-entropy as the loss function. We applied the callback methods that include early stopping, checkpointing the model, as well as reduction of learning rate to prevent overfitting and maximize the output.

Model Performance Metrics

Table 1. Among all the models tested, EfficientNet was shown to be the most efficient, as it achieved higher accuracy and F1-score than other models.

Model	Accuracy (%)	Precision	Reall	F1-score
LeNet	61.89	0.62	0.63	0.62
VGG-16	66.46	0.68	0.70	0.69
ResNet-50	61.68	0.62	0.62	0.62
EfficientNet	**75.93**	**0.86**	**0.86**	**0.86**

We used accuracy, precision, recall, and F1-score for

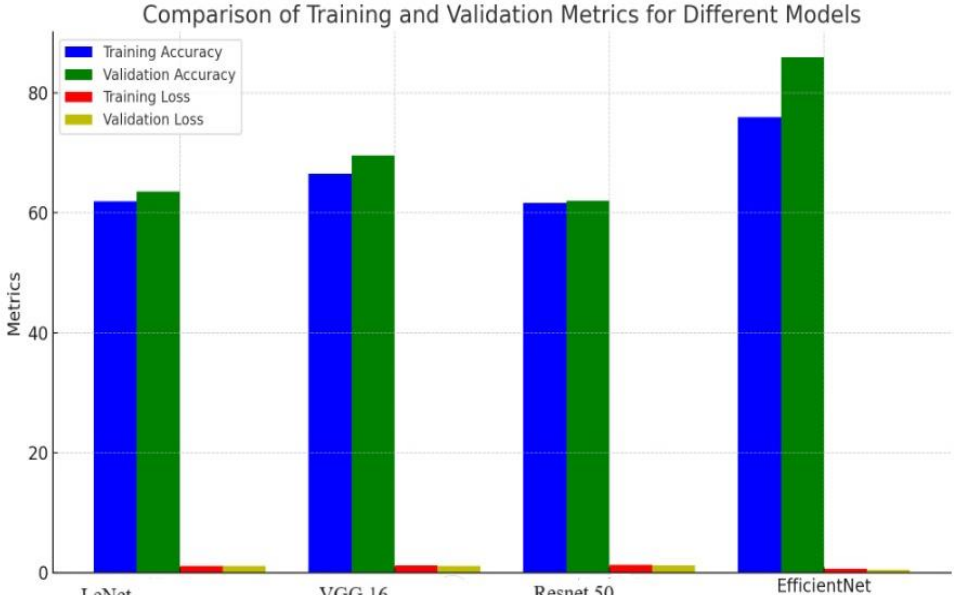


Figure 20. shows a comparison of training and validation metrics of four models.

DOI: https://doi.org/10.52756/ijerr.2025.v47.008

standard performance metrics to measure model effectiveness objectively. Summary of metrics for each model is given in Table 1.

Although CNNs give high accuracy in the automated classification of diseases, traditional inspection methods are the widely used diagnostic methods for cassava diseases. In our experiments, we demonstrate that CNN-based models are as accurate as or even more accurate but much more scalable and efficient, which is especially useful for remote areas where experts may not be available.

When comparing the effectiveness of different DL models like LeNet, VGG16, ResNet-50, and EfficientNet, it becomes clear that EfficientNet is consistently one step ahead of the rest. Although LeNet and VGG16 are the higher-grade models highly accurate, they still have difficulty with the most complicated tasks, and ResNet-50 is the deepest one that handles those tasks better. Nonetheless, EfficientNet is a novel architecture that distributes power along three channels: depth, width, and resolution. It outperforms the others, in terms of accuracy, efficiency, and generalization on all tasks. Thus, EfficientNet is the best option, among the models, for high-performance tasks that need a model.

Conclusion

Crop and plant products, when grown, are essential as they provide us with food, fuel, clothing, and various other things that we need, which are horticulture's mainstays. Guaranteeing the health of such plants is very important because they may be attacked by pests, viruses, or diseases that will reduce their quality and quantity. Plant protection meaning monitoring and removing such threats leads to sustainable agricultural development and hence lessens the stress farmers go through.

At this, the timely and right identification of diseased plants is the most important thing in ensuring agricultural production. However, the methods available for detecting this problem have been used for a long time; this research concentrates on applying deep learning techniques instead. These latest procedures have the potential to take over conventional farming tools with the help of artificial intelligence specialized APIs and intelligent robots which are efficient in correctly recognizing plants.

In our investigation, the four deep learning models, LeNet, VGG16, ResNet-50 and EfficientNet are significant and were tried out. They were trained with sufficient data including images of both healthy and contaminated plants. The aim was to enhance the accuracy of disease recognition which is necessary for early intervention and care. For the examination of the results, we submitted that all the models were quite successful, however, EfficientNet emerged as the best model in view of the speed of execution and the precision in identifying the plant diseases. The value of using advanced models such as these is in the fact that they provide farmers and agricultural professionals with tools for maintaining crop health and thus, maximizing yield.

Future Scope

This study utilized publicly available web resources to compile the dataset; however, future studies will be oriented toward gathering more diverse and representative data directly from the field. Once conditions allow, we plan to capture high-quality images of crops from farms and agricultural fields to create a more comprehensive and robust dataset. This will not only improve the generalizability of the model but also allow for more accurate disease detection in real-world conditions.

Crops are also vulnerable to damage by animals like rats, mice, spiders, and birds, in addition to diseases. By recognizing these complexities, we seek to broaden the range of the model to incorporate the detection of these other threats as well. Our future efforts will involve the incorporation of this improved model into drones which will facilitate real-time monitoring and early detection of a wide range of crop problems. Thus, it is hoped that we can develop a more refined and powerful system that will precisely recognize even the most complicated infections and threats, which in turn will be the key to the resilience of agricultural practices.

This research ultimately opens up the possibility of the usage of advanced machine learning techniques in agriculture and it may be a major breakthrough in the way we think of plant protection and crop management in the future. Moreover, the advancement of our model and the expansion of our model will lead to a future where technology is the central component of global food security and farmers' vital work.

Conflict of Interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

Akash, F. R., Priniya, A. A., Chadni, J. S., Shuha, J. A., Emu, I. A., Reza, A. W., & Arefin, M. S. (2023). Deep Learning Approach for COVID-19 Detection: A Diagnostic Tool Based on VGG16 and VGG19. *Intelligent Computing and Optimization*, 394–403. https://doi.org/10.1007/978-3-031-50158-6_39

Albattah, W., Nawaz, M., Javed, A., Masood, M., & Albahli, S. (2021). A novel deep learning method for detection and classification of plant diseases. *Complex & Intelligent Systems*, *8*(1), 507–524. https://doi.org/10.1007/s40747-021-00536-1

Aruna, R., Prabu, M., Ananthi, S., Bharathi, V. C., Sathya, R., & Suchithra, B. (2023). Vision based Cassava Plant Leaf Disease Classification using Machine Learning Techniques. *2nd International Conference on Automation, Computing and Renewable Systems* (ICACRS), 959–964. https://doi.org/10.1109/icacrs58579.2023.10404468

Azeta, A., Jonathan, K., Guembe, B., & Nwadoziokwu, E. (2023). A Deep Learning Approach for Cassava Leaf Disease Diagnosis. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4669543

Deep Convolution Neural Network Based solution for Detecting Plant Diseases. (2022). *Journal of Pharmaceutical Negative Results*, *13*(S01). https://doi.org/10.47750/pnr.2022.13.s01.57

Demilie, W. B. (2024). Plant disease detection and classification techniques: a comparative study of the performances. *Journal of Big Data*, *11*(1). https://doi.org/10.1186/s40537-023-00863-9

Emmanuel, A., Mwangi, R. W., Murithi, P., Fredrick, K., & Danison, T. (2023). Classification of cassava leaf diseases using deep Gaussian transfer learning model. *Engineering Reports*, *5*(9). Portico. https://doi.org/10.1002/eng2.12651

Jayaprakash, K., & Balamurugan, S. P. (2020). Analysis of Plant Disease Detection and Classification Models: A Computer Vision Perspective. *Journal of Computational and Theoretical Nanoscience*, *17*(12), 5422–5428. https://doi.org/10.1166/jctn.2020.9435

Kumar, A., Singh, S., Gaurav, A. K., Srivastava, S., & Verma, J. P. (2020). Plant Growth-Promoting Bacteria: Biological Tools for the Mitigation of Salinity Stress in Plants. *Frontiers in Microbiology*, *11*. https://doi.org/10.3389/fmicb.2020.01216

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Li, R., Chen, S., Matsumoto, H., Gouda, M., Gafforov, Y., Wang, M., & Liu, Y. (2023). Predicting rice diseases using advanced technologies at different scales: present status and future perspectives. *Abiotech.*, *4*(4), 359–371. https://doi.org/10.1007/s42994-023-00126-4

Liang, C., Amelung, W., Lehmann, J., & Kästner, M. (2019). Quantitative assessment of microbial necromass contribution to soil organic matter. *Global Change Biology*, *25*(11), 3578–3590. Portico. https://doi.org/10.1111/gcb.14781

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*, *7*. https://doi.org/10.3389/fpls.2016.01419

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*, *7*. https://doi.org/10.3389/fpls.2016.01419

Pathak, Y., Shukla, P. K., Tiwari, A., Stalin, S., Singh, S., & Shukla, P. K. (2022). Deep Transfer Learning Based Classification Model for COVID-19 Disease. *IRBM*, *43*(2), 87–92. https://doi.org/10.1016/j.irbm.2020.05.003

Prasad, P., & Agniraj, S. (2024). Cross-Domain Adaptation Techniques for Robust Plant Disease Detection: A DANN-CORAL Hybrid Approach. *International Journal of Experimental Research and Review*, *42*, 68-84. https://doi.org/10.52756/ijerr.2024.v42.007

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., & Prabhat. (2019). Deep learning and process understanding for data-driven Earth system science. *Nature*, *566*(7743), 195–204. https://doi.org/10.1038/s41586-019-0912-1

Shruthi, U., Nagaveni, V., & Raghavendra, B. K. (2019). A Review on Machine Learning Classification Techniques for Plant Disease Detection. *5th International Conference on Advanced Computing & Communication Systems* (ICACCS), 281–284. https://doi.org/10.1109/icaccs.2019.8728415

Shukla, S., Upadhyay, D., Mishra, A., Jindal, T., & Shukla, K. (2022). Challenges Faced by Farmers in Crops Production Due to Fungal Pathogens and Their Effect on Indian Economy. *Fungal Diversity, Ecology and Control Management*, 495–505. https://doi.org/10.1007/978-981-16-8877-5_24

Thakur, P. S., Khanna, P., Sheorey, T., & Ojha, A. (2022). Trends in vision-based machine learning techniques for plant disease identification: A systematic review. *Expert Systems with Applications*, *208*, 118117. https://doi.org/10.1016/j.eswa.2022.118117

Uke, A., Tokunaga, H., Utsumi, Y., Vu, N. A., Nhan, P. T., Srean, P., Hy, N. H., Ham, L. H., Lopez-Lavalle, L. A. B., Ishitani, M., Hung, N., Tuan, L. N., Van Hong, N., Huy, N. Q., Hoat, T. X., Takasu, K., Seki, M., & Ugaki, M. (2021). Cassava mosaic disease and its management in Southeast Asia. *Plant Molecular Biology*, *109*(3), 301–311. https://doi.org/10.1007/s11103-021-01168-2

Upadhyay, S., Jain, J., & Prasad, R. (2024). Early Blight and Late Blight Disease Detection in Potato Using Efficientnetb0. *International Journal of Experimental Research and Review*, *38*, 15-25. https://doi.org/10.52756/ijerr.2024.v38.002

Vardhan, J.V., & Swetha, K.S. (2023). Detection of healthy and diseased crops in drone captured images using Deep Learning. *ArXiv*, *abs/2305.13490*.

Wu, S. Z., Al-Eryani, G., Roden, D. L., Junankar, S., Harvey, K., Andersson, A., Thennavan, A., Wang, C., Torpy, J. R., Bartonicek, N., Wang, T., Larsson, L., Kaczorowski, D., Weisenfeld, N. I., Uytingco, C. R., Chew, J. G., Bent, Z. W., Chan, C.-L., Gnanasambandapillai, V., … Swarbrick, A. (2021). A single-cell and spatially resolved atlas of human breast cancers. *Nature Genetics*, *53*(9), 1334–1347. https://doi.org/10.1038/s41588-021-00911-1

Wulder, M., White, J., Hermosilla, T., & Coops, N. (2021). Air-and spaceborne lidar data as reference for satellite-based forest parameter retrieval and mapping over space and time. September, Silvilaser 2021, Vienna, Austria.

How to cite this Article:

Poonam Chaudhary, Monika Lamba and Sneha K (2025). Enhancing Cassava Disease Detection: Leveraging Deep Convolutional Neural Networks and Data Augmentation for Accurate Diagnosis. *International Journal of Experimental Research and Review*, *47*, 93-107.

DOI : https://doi.org/10.52756/ijerr.2025.v47.008

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.